

Заняття 1.

Частина 1. Python



Мова програмування python отримала свою назву завдяки комедійному телешоу “Літаючий цирк Монті Пайтон”, хоча традиційно, в російськомовного населення, ця мова програмування асоціюється з пітоном.

Завдяки деяким особливостям Python відмінно підходить для новачків. Головне - на ньому можна писати прості та ефективні програми, не витрачаючи на це багато часу. В Python використовується менше складних спеціальних символів, ніж в більшості інших мов, так що програми на ньому легко читаються. Хоча, разом з тим Python використовується і для великих серйозних програм. Мова програмування використовується для створення веб-сайтів, ігор, програм та штучного інтелекту.

Давай привітаємось



Насправді, комп'ютер абсолютно нічого не вміє і не знає. Все що він розуміє - це 0 і 1, тобто пройшов сигнал чи ні. І те що комп'ютер таки може виконувати забезпечується програмним забезпеченням, яке придумали і створили люди. І сьогодні ми почнемо також навчати комп'ютер. І першим нашим кроком буде привітання.

Подивіться уважно на верхній скріншот на слайді. На ньому написана команда. Як ви думаєте що вона означає?

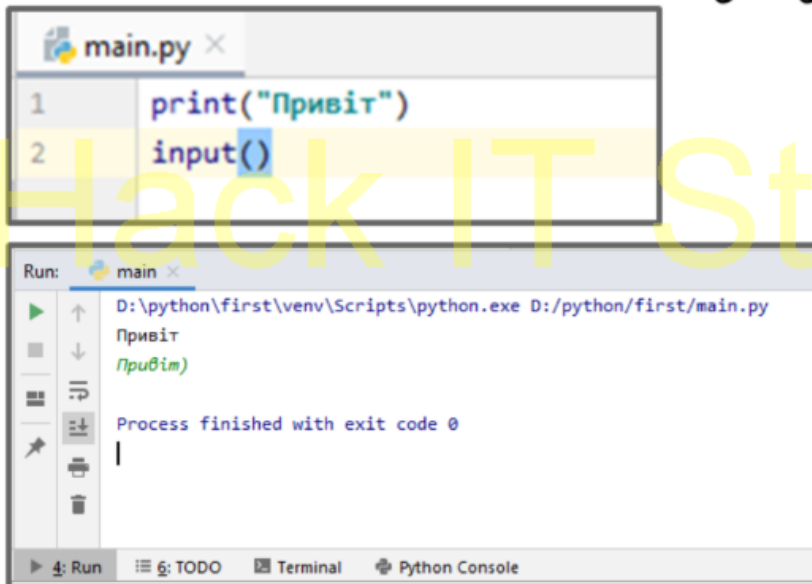
Звичайно, як ви могли здогадатись, це команда друкування, тобто виведення на екран. Те що ми хочемо вивести записується в круглі дужки, які слідує відразу за назвою команди, без пробілів.

Привіт - це у нас просто фраза, текст, який міг би бути будь яким іншим. Щоб комп'ютер не спутав звичайний текст із англійськими словами, які означають команди, текст завжди виділяється лапками, як це показано на слайді.

Отож давайте запишемо цю команду до нашої програми і навчимо комп'ютер з нами вітатись.

Як ми бачимо, тепер у нас на екрані виведення з'явилося слово Привіт і результат виконання з кодом 0, тобто комп'ютер привітався з нами і сказав що програма виконалась без помилок.

Зворотній зв'язок



```
main.py x
1 print("Привіт")
2 input()

Run: main x
D:\python\first\venv\Scripts\python.exe D:/python/first/main.py
Привіт
Привіт)
Process finished with exit code 0
```



Для того щоб між нами і комп'ютером почався діалог, ми також повинні щось говорити комп'ютеру, а команда, яка дозволяє йому нас слухати, називається `input`.

Можливо хтось знає англійську і може перекласти?

Так це два слова `in` - в і `put` класти, тобто ніби ми щось кладемо в комп'ютер. Використання цієї програми ви можете побачити на верхньому скріншоті в другому рядку. Як і в першому випадку, після назви команди у нас слідують круглі дужки. Так як ми нічого туди не "кладемо", то при запуску програми, комп'ютер буде очікувати введення із клавіатури.

Давайте добавимо цю команду в код нашої сторінки і запустимо програму.

У вікні виконання знову з'явилося "Привіт", але у нас немає рядка завершення виконання програми, тому що комп'ютер чекає доки ми теж щось йому "скажемо". Поставте курсор під повідомленням комп'ютера і напишіть привітання йому у відповідь, після чого натисніть ентер.

Тепер програма завершилась.

Інтерпретатор та компілятор



І комп'ютер, як ми вже говорили, насправді не розуміє тих команд, які ми пишемо в ручарм. Він знає лише 0 і 1, а ті команди які ми задаємо, переводять для нього інтерпритатор і компілятор на зрозумілу для нього мову, розписуючи більш детально що потрібно зробити, як в прикладі з пральною машиною. Мова на якій пишемо команди ми, називається вищою мовою програмування, а та, яку розуміє комп'ютер - нижчою. Тобто інтерпритатор і компілятор переводять програму із мови програмування вищого рівня у мову програмування нижчого рівня.

Приклади застосування функції **print**

```
print("Hello, World!")
```

```
>>> print("Hello, World!")
Hello, World!
>>>
```

```
print('tutorial','on','python','print','function')
```

```
>>> print('tutorial','on','python','print','function')
tutorial on python print function
>>> █
```

Увага наступна команда **print**, відрізняється від попередньої. При об'єднанні (додаванні), всі дані мають бути одного типу

```
print('tutorial' + 'on' + 'python' + 'print' + 'function')
```

```
>>> print('tutorial'+ 'on'+ 'python'+ 'print'+ 'function')
tutorialonpythonprintfunction
>>>
```

```
print('tutorial','on','python','print','function',sep='\n')
```

```
>>> print('tutorial','on','python','print','function',sep='\n')
tutorial
on
python
print
function
>>>
```

де **sep** це розділювач між значеннями, може бути будь який дозволений символ чи вираз.

```
print('tutorial','on','python','print','function',sep=',')
```

```
a = 2
```

```
b = "Datacamp"
```

```
print(a,"is an integer while",b,"is a string.")
```

```
a = 2
b = "Datacamp"
print("%d is an integer while %s is a string." %(a,b))
```

```
>>> print("%d is an integer while %s is a string." % (a,b))
2 is an integer while Datacamp is a string.
>>> 
```

```
a = 2
b = "Datacamp"
```

```
print(f"{a} is an integer while {b} is a string.")
```

або

```
print("{0} is an integer while {1} is a string.".format(a,b))
```

Приклади застосування функції **input**

```
name = input()
```

```
name = input("Enter name: ")
```


Завдання *

Використовуючи команди `print()` та `input()`, напишіть свій діалог з комп'ютером



Додаткове завдання, яке виконується учнями, якщо залишився час.